What is claimed is:

1. A method of maintaining cache coherency comprising:

executing a store operation that hits a cache line in a second cache; and

placing the cache line in the second cache in an enhanced exclusive state in response to the store operation, the enhanced exclusive state indicating a copy of the cache line is in a first cache in a modified state.

2. A method as defined in claim 1, wherein the first cache comprises an L1 cache and the second cache comprises an L2 cache.

3. A method as defined in claim 1, further comprising:

placing the cache line in the second cache in an enhanced modified state, the enhanced modified state indicating the copy of the cache line may be in the first cache.

4. A computing apparatus comprising:

a first cache memory;

a second cache memory;

a processor operatively coupled to the first cache memory and the second cache memory, the processor executing a store operation that hits a cache line in the second cache memory, the second cache memory placing the cache line in an enhanced exclusive state in response to the store operation, the enhanced exclusive state indicating a copy of the cache line is in the first cache memory in a modified

state.

5. An apparatus as defined in claim 4, wherein the first cache memory comprises an L1 cache and the second cache memory comprises an L2 cache.

6. An apparatus as defined in claim 4, wherein the second cache memory places the cache line in an enhanced modified state, the enhanced modified state indicating an updated copy of the cache line may be in the first cache memory.

7. A method of victimizing a cache line in a second cache, the method comprising:

issuing an inquiry to a first cache if the cache line is in a predetermined state;

receiving a response to the inquiry if the cache line is in the predetermined state; and

victimizing the cache line in the second cache without a write-back of the cache line from the second cache if the response is indicative of a cache hit.

8. A method as defined in claim 7, wherein the first cache is an L1 cache and the second cache is an L2 cache.

9. A method as defined in claim 7, wherein the predefined state comprises an enhanced modified state indicating a copy of the cache line may be in

the first cache.

10. A method as defined in claim 9, further comprising performing a write-back of the cache line from the second cache without issuing the inquiry to the first cache if the cache line is in a non-enhanced modified state.

11. A method as defined in claim 7, wherein the inquiry comprises an internal inquiry.

12. A method as defined in claim 7, further comprising performing a write-back of the cache line from the second cache if the response is indicative of a cache miss.

13. A method as defined in claim 12, further comprising victimizing the cache line in the second cache after performing the write-back if the response is indicative of a cache miss.

14. A method as defined in claim 7, further comprising victimizing the cache line in the second cache without issuing the inquiry to the first cache if the cache line is in an enhanced exclusive state indicating a copy of the cache line is in the first cache.

15. A method as defined in claim 14, wherein victimizing the cache line in the second cache comprises victimizing the cache line in the second cache without

performing a write-back of the cache line from the second cache.

16. A method as defined in claim 15, wherein the enhanced exclusive state indicates the copy of the cache line in the first cache is in a modified state.

17. An apparatus comprising:

a first cache memory; and

a second cache memory operatively coupled to the first cache memory, the second cache memory being structured to issue an inquiry to the first cache memory if a cache line to be victimized in the second cache memory is in an enhanced modified state, the enhanced modified state indicating a copy of the cache line may be in the first cache memory.

18. An apparatus as defined in claim 17, wherein the first cache memory is an L1 cache and the second cache memory is an L2 cache.

19. An apparatus as defined in claim 17, further comprising a main memory operatively coupled to the second cache memory, wherein the second cache memory is structured to:

receive a response to the inquiry; and

victimize the cache line in the second cache memory without a write-back of the cache line to the main memory if the response is indicative of a cache hit.

20. An apparatus as defined in claim 19, wherein the second cache memory is structured to perform the write-back of the cache line without issuing the inquiry to the first cache memory if the cache line is not in the enhanced modified state.

21. An apparatus as defined in claim 19, wherein the second cache memory performs the write-back of the cache line if the response is indicative of a cache miss.

22. An apparatus as defined in claim 19, wherein the second cache memory is structured to victimize the cache line in the second cache after performing the write-back if the response is indicative of a cache miss.

23. An apparatus as defined in claim 17, wherein the second cache memory is structured to victimize the cache line in the second cache memory without issuing the inquiry to the first cache memory if the cache line is in an enhanced exclusive state indicating a copy of the cache line is in the first cache memory in a modified state.

24. A method of performing a cache snoop, the method comprising:

posting a snoop hit signal if a cache line in a second cache is in one of an exclusive state, an enhanced exclusive state, and a shared state, wherein the enhanced exclusive state indicates a copy of the cache line is in a first cache; and

posting a first snoop hit-modified signal if the cache line in the second

cache is in one of a modified state and an enhanced modified state, wherein the enhanced modified state indicates an updated copy of the cache line may be in the first cache.

25. A method as defined in claim 24, wherein the first cache comprises an L1 cache and the second cache comprises an L2 cache.

26. A method as defined in claim 24, further comprising:

detecting a second snoop hit-modified signal from the first cache; and

invalidating the cache line in the second cache in response to detecting the second snoop hit-modified signal from the first cache.

27. A method as defined in claim 26, wherein detecting the second snoop hit-modified signal from the first cache implies a write-back of the cache line from the first cache.

28. A method as defined in claim 24, further comprising detecting an absence of a second snoop hit-modified signal from the first cache, wherein the absence of the second snoop hit-modified signal from the first cache implies a write-back of the cache line from the second cache.

29. An apparatus for performing a cache snoop, the apparatus comprising:

a processor;

a first cache operatively coupled to the processor; and

a second cache operatively coupled to the processor, the second cache being structured to post a snoop hit signal if a cache line in the second cache is in one of an exclusive state, an enhanced exclusive state, and a shared state, wherein the enhanced exclusive state indicates a modified copy of the cache line is in the first cache.

30. An apparatus as defined in claim 29, wherein the second cache is structured to post a first snoop hit-modified signal if the cache line in the second cache is in one of a modified state and an enhanced modified state, wherein the enhanced modified state indicates an updated copy of the cache line may be in the first cache.

31. An apparatus as defined in claim 29, wherein the first cache comprises an L1 cache and the second cache comprises an L2 cache.

32. An apparatus as defined in claim 29, wherein the second cache is structured to:

detect a second snoop hit-modified signal from the first cache; and

invalidate the cache line in the second cache in response to detecting the second snoop hit-modified signal from the first cache.